Topological Lines in 3D Tensor Fields

Xiaoqiang Zheng and Alex Pang
Computer Science Department
University of California, Santa Cruz, CA 95064
zhengxq@cse.ucsc.edu, pang@cse.ucsc.edu

Abstract

Visualization of 3D tensor fields continues to be a major challenge in terms of providing intuitive and uncluttered images that allow the users to better understand their data. The primary focus of this paper is on finding a formulation that lends itself to a stable numerical algorithm for extracting stable and persistent topological features from 2nd order real symmetric 3D tensors. While features in 2D tensors can be identified as either wedge or trisector points, in 3D, the corresponding stable features are lines, not just points. These topological feature lines provide a compact representation of the 3D tensor field and are essential in helping scientists and engineers understand their complex nature. Existing techniques work by finding degenerate points and are not numerically stable, and worse, produce both false positive and false negative feature points. This paper seeks to address this problem with a robust algorithm that can extract these features in a numerically stable, accurate, and complete manner.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques;

Keywords: hyperstreamlines, real symmetric tensors, degenerate tensors, tensor topology, topological lines

1 Introduction

The main motivation and goal of this paper is to develop a simple yet powerful representation of 3D real symmetric tensor fields. Topology-based methods prove to yield simplified yet effective representation in many visualization fields. The topological structures are simple for users to understand the underlying data fields yet sensitive enough to capture important features. In most cases, trained users can even reconstruct the data fields by looking at the topological structures. Early work on using topology-based method to visualize tensor fields by [5, 7] lays an important background for this research project. It defines the tensor topology based on degenerate features and discusses its nature in 2D cases in great details, and provides useful knowledge in 3D cases. But we find this early work insufficient in studying 3D tensor topology. Not only is the dimension of the features unknown, but how to numerically extract the topological structures is also obscure. In their previous work, Hesselink mentioned that the dimension of the degenerate features can be points, lines, surfaces or subvolumes. This claim itself is essentially true, but it does not point out the dimension of features in 3D topology in general, i.e., in a typical non-degenerate data. By analogy, although the critical features in 3D vector fields can be lines, surfaces or even subvolumes, we know they are mostly points in typical non-degenerate data. This knowledge is the foundation of the study of topological structure in vector visualization. All the subsequent study on separatrices and other topological features are based on the extraction of the critical *points*. On the other hand, no topological results on 3D real symmetric tensor fields been published to date indicating that topological tensor features form *lines*.

During our research on 3D tensor topology, we confirmed that the topological structures in 3D real symmetric tensor fields form feature lines. This can be verified by an early theorem pointed out by Wigner and von Neumann that the real symmetric degenerate matrices form a variety of codimension two [14]. This discovery is important in that it tells us that future studies on topology-based method for 3D non-degenerate real symmetric tensor fields should be based on feature *lines*. We can capture important features, study the underlying tensor fields and even reconstruct the data fields by looking at these extracted topological structural lines with their separatrix surfaces.

Traditionally, the degenerate features in 3D tensor fields are defined as tensors whose cubic discriminant is equal to zero. Finding roots of discriminants in a stable manner proves to be a challenging task because of their high-orderedness and singularity. In this paper, we find an alternative formulation that decomposes the cubic discriminant into the sum of the squares of seven cubic polynomials referred to as *discriminant constraint functions*. Through this decomposition, the tensors whose discriminant equals zero are equivalent to the tensor whose individual tensor constraint functions all equal zero at the same time. This formulation eliminates the high-orderedness and singularity problems encountered in existing practice, and thus makes developing stable numerical algorithm to extract degenerate feature lines possible.

2 Tensor Analysis

Tensor fields, especially second-order tensor fields, are useful in many medical, mechanical and physical applications such as: fluid dynamics, meteorology, molecular dynamics, biology, astrophysics, mechanics, material science and earth science. Effective tensor visualization methods can enhance research in a wide variety of fields. However, developing an effective algorithm can be difficult because of the large amount of information contained in 3D tensor fields: there are nine independent components in each tensor and six for a symmetric tensor. Users in many research fields are especially interested in real symmetric tensors. In some applications, the data themselves are inherently symmetric. In other cases, symmetric tensor data can be obtained through various decomposition techniques.

2.1 Degenerate Tensors and Discriminants

Each real symmetric tensor can be decomposed into three orthogonal eigenvectors, each of which has an eigenvalue associated with it. They are labeled as major, medium and minor eigenvectors according to the relative magnitudes of their eigenvalues. In non-degenerate cases, these eigenvectors do not cross each other. The degenerate features are then defined as those where the eigenvectors could cross each other. Hesselink et al. show that the only degenerate features are those having at least two equal eigenvalues [7]. Fortunately, we do not need to conduct the eigen decomposition to

IEEE Visualization 2004
October 10-15, Austin, Texas, USA
0-7803-8788-0/04/$20.00 ©2004 IEEE

find the degenerate points. A tensor has two (or three) equal eigenvalues if and only if its discriminant equals zero. The discriminant D_3 of a tensor T with eigenvalues λ_1, λ_2 and λ_3 is defined as,

$$T = \begin{pmatrix} T_{00} & T_{01} & T_{02} \\ T_{01} & T_{11} & T_{12} \\ T_{02} & T_{12} & T_{22} \end{pmatrix} \quad (1)$$

$$D_3(T) = (\lambda_1 - \lambda_2)^2(\lambda_2 - \lambda_3)^2(\lambda_3 - \lambda_1)^2 \quad (2)$$

This can be reformulated into a form that does not require eigen decomposition to explicitly determine eigenvalues as follows:

$$P = T_{00} + T_{11} + T_{22} \quad (3)$$

$$Q = \begin{vmatrix} T_{00} & T_{01} \\ T_{01} & T_{11} \end{vmatrix} + \begin{vmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{vmatrix} + \begin{vmatrix} T_{22} & T_{02} \\ T_{02} & T_{00} \end{vmatrix} \quad (4)$$

$$R = \begin{vmatrix} T_{00} & T_{01} & T_{02} \\ T_{01} & T_{11} & T_{12} \\ T_{02} & T_{12} & T_{22} \end{vmatrix} \quad (5)$$

$$D_3(T) = Q^2P^2 - 4RP^3 - 4Q^3 + 18PQR - 27R^2 \quad (6)$$

From Equation 2, we can easily find that a discriminant is (a) always non-negative; (b) equal to zero if and only if at least two of the eigenvalues are equal. And it is always perfect for computation and numerical purposes because although it is defined on eigenvalues, we do not really need to carry out an expensive eigen decomposition. Instead, we only need to compute Equation 6 which is a polynomial of order six to get the discriminant.

The justification to define tensors with two or more equal eigenvalues as the degenerate features is simple. In a vector field, the streamline integration is ambiguous at a point with zero velocity. In a tensor field, the hyperstreamline integration is also ambiguous at a point with two equal eigenvalues, because any linear combination of the two eigenvectors is another eigenvector. Since the degenerate tensors are the only places where hyperstreamlines can cross each other, they play an important role in 3D tensor field topological analysis, which divides the space into smaller subspaces such that local hyperstreamlines exhibit similar patterns. In general, the degenerate tensors can be interpreted in a similar fashion as the critical points in vector field visualization.

2.2 Dimensional Analysis

A feature set F is a subset of space S with degree of freedom s and is defined on a domain with dimension D. A point P belongs to F if and only if a set of constraints C are satisfied i.e., $C(P) = 0$, where there are c independent equations in C. And if when $C(P) = 0$, R degrees of freedom in P become indistinguishable, where we define that there are R degrees of redundancy in the constraints C. Then in a non-degenerate case, the dimension of the feature F on domain with dimension D is H and is defined as:

$$H = D - C - R \quad (7)$$

If we already know that the feature set F forms a space with a dimension of G, then the dimension is:

$$H = D + G - S \quad (8)$$

That a type of feature is of dimension $H \geq 0$ means it stays mostly *stable* in a form of dimension H for non-degenerate data sets. Although they can also be in forms of other dimensions, such as in cases of unstable features, their existence can be disturbed by small errors introduced by computational or even interpolation methods. Therefore, we define a feature to be *stable* only if the *existence* of the feature is *not* affected by the presence of a small noise – even if the introduction of noise alters the location and properties of the feature. For example, if the intersection point of two lines on a plane is defined as a feature, then the intersection point of two straight lines is simply stable. For the case where the two straight lines lie right on top of each other, the feature turns into a line. However, this case is not stable because although all the points on these overlapping lines are feature points, a small disturbance in the positions or orientations of these two lines will reduce the feature line back to a single feature point. We use this type of dimensional analysis to get our first idea of the dimension of stable features in degenerate and non-degenerate 3D tensors.

3 Previous Work

Early tensor visualization techniques relied on the tensor ellipsoid which deformed a spherical glyph according to the eigenvalues of the tensors. Variations of the basic tensor ellipsoid include drawing eigenvalue scaled axes for the eigenvectors, Haber's disk and rod glyphs [6], flow probe [3], Laidlaw et al. glyphs using brush strokes [12, 11]. More recently, Gordon Kindlmann proposed new glyphs that uses superquadric tensor glyphs to visualize tensors as a combination of spherical, planar and linear tensors [9]. With few exceptions, particularly for the case of 3D tensor fields, glyphs are used in a sparing manner because of the clutter and occlusion they produce. Hence, they provide a discrete, rather than a continuous view of the tensor field. To address this problem, tensor splats were introduced by [1] to provide a global continuous view of the tensor field. Using a barycentric mapping of linear, planar, and spherical tensors, different parts of the tensor volume can be highlighted.

A hyperstreamline is basically a streamline defined over an eigenvector field [4]. Typically, the major eigenvector field is used for integrating the hyperstreamline, while the two other eigenvector fields provide local information along the length of the major hyperstreamline and are mapped to its cross section. One of the weakness of hyperstreamlines is ambiguity in places where the tensors are isotropic or even planar, i.e. the eigenvalues are nearly equal. In these areas, a sudden change in direction of the hyperstreamline may arise. To address this problem, tensorlines were introduced by [18]. Ambiguities are resolved by taking the anisotropy of the local tensor into account as well as information about orientation of nearby features. This allows the tensorlines to proceed in a relatively smooth path, even in the face of isotropic regions or noise in the data set.

Topology-based tensor visualization techniques represent the tensor fields in a simple yet powerful way. The critical features are extracted as a simplified version of the underlying data field. Experienced users can understand and even reconstruct the tensor data by looking at their properties and patterns. The critical feature is defined as degenerate tensors where the eigenvalues are identical, and are the only places that the two associated hyperstreamlines can intersect themselves. In 2D tensor fields, there is only one way to obtain a degenerate point: the two eigenvalues must be equal. Hesselink and Delmarcelle used this concept in 2D and discussed the nature of the degenerate points (wedges and trisectors) in great detail. However, it is less successful in 3D, in part because there are two types of degenerate points in 3D: double and triple degenerate points, where two of the three eigenvalues are equal, and all three eigenvalues are identical, respectively. This early work does not fully explore the properties of the double degenerate features and instead focuses on the triple degenerate tensors, whose properties are closer to their counterparts in 2D. In [7], they hint that the triple degenerate points (for the double point load data) are connected by a locus of double degenerate points. The paper fails to point out that the dimension of the stable double degenerate features are in fact lines in most of the typical non-degenerate tensor fields. And hence, it did not talk about finding a stable numerical method to extract these feature lines in 3D.

Although triple degenerate features are useful, they are extremely rare and unstable. Using the *dimensional analysis* technique, we know the dimensions of triple degenerate features in 3D is of dimension $H = D + G - S = 3 + 1 - 6 = -2$. $G = 1$ is due to the fact that the triple degenerate tensors are only an identity tensor times a constant, so that they have only one degree of freedom. $S = 6$ is because a real symmetric tensor has six independent components, hence it has six degrees of freedoms. A dimensionality that is less than zero means the feature is unstable. In other words, not only is it extremely rare in real data, but its very existence will also be easily dissolved by small errors introduced by numerical and interpolation methods. This property dramatically limits the usefulness of triple degenerate tensors in practical contexts. Even in time-varying data, which has a dimension of four, the dimensions of the feature of $H = 4 + 1 - 6 = -1$, is still unstable. Actually, in our experiments with many real time-varying stress datasets, we have not found any triple degenerate points. The only data set that contains triple degenerate points is from a synthetic data set known as the Boussinesq double point load stress tensor.

In complex 2D tensor fields, the extracted topology may also be very complex. [17] proposed algorithms to simplify 2D tensor topology as well as track them in time-varying 2D tensor fields [16].

Anisotropy in tensor fields are important features in some fields such as diffusion tensor MRI. In 3D, tensors are classified as being linear or anisotropic where there is a predominant eigenvalue and two other smaller eigenvalues, planar where there are two roughly equal eigenvalues and one smaller one, and spherical or isotropic where there are three roughly identical eigenvalues. [10] uses barycentric coordinates to map these properties to color and opacity in volume rendering tensor fields. An alternative approach, called HyperLIC, was proposed by [20] to highlight anisotropy using textures. Linear tensors are represented by highly correlated, high contrast textures, while isotropic tensors end up as blurry textures with no preferred orientation.

Using the physical analogy of bending steel beams under load, [2] used deformation to visualize the effects of tensor fields. Idealized objects such as lines, surfaces and subvolumes are deformed under tensor transformations. This was further improved by [19] to provide a globally consistent deformation based on a collection of local deformations. Calculations were carried using a system of springs.

Extending the idea of deformation to optics, [21] added three alternative ways of visualizing tensors: (a) First, light rays were traced through a tensor volume and bent according to the local tensor properties that they encounter. The bent rays show divergent or convergent regions in the tensor field. (b) Second, the exit points of the rays are collected, as in caustic ray tracing. Different wavelengths are simulated and color separation on the resulting caustic image provides a dense visualization of divergence and convergence from a given viewpoint. (c) Third, the tensor field is treated as a lens that distorts an image. Studying the distortion of a known image, e.g. checkerboard pattern, reveals compressive and tensile regions in the tensor field.

4 Methods

In this section, we first show that the features are indeed lines in non-degenerate data set. Then we discuss the details of our numerical methods to extract the degenerate features in 3D real symmetric tensor fields.

4.1 Dimensionality of Tensor Features

Before we can extract the critical features from 3D tensor fields, we need to know what kind of features we are looking for. Algorithms to locate points, lines, surfaces and volumes employ very different strategies. As mentioned earlier, we found that for most non-degenerate 3D tensors, the dimensionality of the critical feature is *one* and hence they form feature lines. This can be shown using dimensional analysis described earlier, or with the theorem by von Neumann and Wigner which states that the real symmetric degenerate matrices form a variety of codimension two [14]. Codimension is defined as the difference of the dimensions between a subspace and the space where it is defined. An interpretation of this theorem is that one of the two codimensions is introduced by the constraint that at least two eigenvalues must be equal. The other codimension is introduced implicitly by the fact that when two eigenvalues are equal, the associated eigenvectors are undetermined up to one degree of freedom, since any orthogonal linear combinations of the eigenvectors yield another valid pair. For real symmetric 3D tensors, we have six degrees of freedom. Hence, it also follows from this theorem that these tensors form a variety of dimension four. An approach to parameterize the 3D degenerate tensors using four parameters is introduced in [22].

For a tensor field with a spatial dimension of $D = 3$, the constraint that two eigenvalues are equal is a constraint with a dimension of $C = 1$. In the meantime, the eigenvectors have $R = 1$ degree of redundancy as shown above. Thus the dimension of the features is: $H = D - C - R = 3 - 1 - 1 = 1$. That means, for a non-degenerate tensor field, the degenerate features are lines. Using the theorem by von Neumann and Wigner, we see that dimension $D = 3$ and codimension is 2, so the features in the data form a subspace of dimension $3 - 2 = 1$, i.e. lines. While the main features are lines, it is still possible to have features that are points, surfaces or subvolumes, but those types of features would be considered unstable and do not persist. Those types of unstable features are also less common in most 3D tensor fields. As such, we focus our tensor feature extraction to extract feature lines rather than surfaces or subvolumes. We still need to extract points as these form the basis for the feature lines. Because of this design criterion, features that are surfaces (e.g. in the single point load data) or subvolumes may not be detected as readily as feature lines. This limitation is not insurmountable, but is rather based on the effective use of limited resources in finding features that are not as common nor stable.

4.2 Constraint Functions

To find the critical degenerate tensors, we need to locate those tensors whose discriminants are zero. Although Equation 6 provides an elegant representation for evaluating the discriminant without having to perform eigen decomposition, it is not very suitable for finding roots. In Equation 6, the discriminant of a real symmetric tensor is a polynomial of order six. Since it is always non-negative, the degenerate tensor also happens to be its minimum. A good method widely used to find the root of an equation is to detect the change of signs and then to recursively bisect the domain of interest. But because the degenerate feature is itself a minimum, there is no change of sign at all. Relying on the gradients is also dangerous, because the gradients are notoriously unstable unless they are very close to the feature. Due to this high-orderedness and singularity, directly finding the root of a cubic discriminant stably is very difficult. Instead, we look for another representation of the discriminant.

In our investigation so far, we found that while [8] pointed out that not all non-negative polynomials can be broken down into the sum of squares of polynomials, the cubic discriminant can be written as the sum of the squares of *seven* polynomials. We also learned that not only can the discriminant of a second-order tensor of any dimension be expressed as the sum of squares [13], but our solution to the 3D case of *seven* equations is optimal [15]. Therefore, the definition of degenerate tensors can also be expressed as the tensors where the seven *discriminant constraint functions* are all zero at the same time. We use these seven cubic equations to extract the feature lines from 3D tensor fields. The seven discriminant constraints are:

$$\begin{aligned}
f_x(T) = \quad & T_{00}(T_{11}^2 - T_{22}^2) + T_{00}(T_{01}^2 - T_{02}^2) + T_{11}(T_{22}^2 - T_{00}^2) + \\
& T_{11}(T_{12}^2 - T_{01}^2) + T_{22}(T_{00}^2 - T_{11}^2) + T_{22}(T_{02}^2 - T_{12}^2) \\
f_{y1}(T) = \quad & T_{12}(2(T_{12}^2 - T_{00}^2) - (T_{02}^2 + T_{01}^2) + 2(T_{11}T_{00} + T_{22}T_{00} \\
& -T_{11}T_{22})) + T_{01}T_{02}(2T_{00} - T_{22} - T_{11}) \\
f_{y2}(T) = \quad & T_{02}(2(T_{02}^2 - T_{11}^2) - (T_{01}^2 + T_{12}^2) + 2(T_{22}T_{11} + T_{00}T_{11} \\
& -T_{22}T_{00})) + T_{12}T_{01}(2T_{11} - T_{00} - T_{22}) \\
f_{y3}(T) = \quad & T_{01}(2(T_{01}^2 - T_{22}^2) - (T_{12}^2 + T_{02}^2) + 2(T_{00}T_{22} + T_{11}T_{22} \\
& -T_{00}T_{11})) + T_{02}T_{12}(2T_{22} - T_{11} - T_{00}) \\
f_{z1}(T) = \quad & T_{12}(T_{02}^2 - T_{01}^2) + T_{01}T_{02}(T_{11} - T_{22}) \\
f_{z2}(T) = \quad & T_{02}(T_{01}^2 - T_{12}^2) + T_{12}T_{01}(T_{22} - T_{00}) \\
f_{z3}(T) = \quad & T_{01}(T_{12}^2 - T_{02}^2) + T_{02}T_{12}(T_{00} - T_{11})
\end{aligned}$$

$$\begin{aligned}
D_3(T) = \quad & f_x(T)^2 + f_{y1}(T)^2 + f_{y2}(T)^2 + f_{y3}(T)^2 + \\
& 15f_{z1}(T)^2 + 15f_{z2}(T)^2 + 15f_{z3}(T)^2
\end{aligned} \tag{9}$$

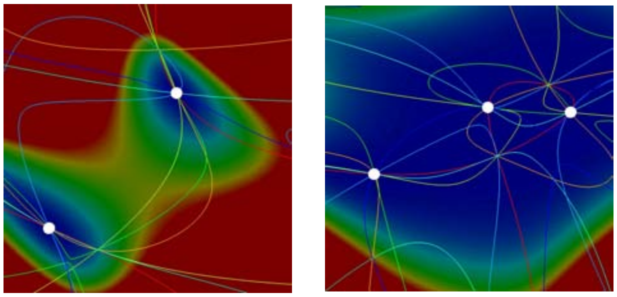
A tensor is degenerate if and only if all of its seven discriminant constraint functions are zero. This is the condition that we employ to extract the critical features in 3D tensor fields. Its first advantage is that the constraint functions are only cubic polynomials, instead of a polynomial of order of six which tend to oscillate more. This property leads to a more stable and accurate numerical algorithm. In addition, the requirement that all seven constraint functions be zero at the same time depends on the tensor value only and not on the gradient calculated from adjacent tensors. Hence, the algorithm yields a more accurate result than the algorithms that rely on finding critical points where the gradients of the discriminants are zeros. Its second advantage is that the constraint functions can be both positive or negative, as opposed to always being non-negative. This property allows us to perform a fast and inexpensive check for the existence of features. And finally, the reformulation also does not require eigen decomposition.

Although degenerate features require that all seven discriminant constraint functions be zero at the same time, *pseudo-features* with less than seven constraints satisfying this criterion are also very interesting (See Figure 1). In our experiments, some of these pseudo-feature points have three to five constraints as zeros. Our study into them reveals some other interesting properties of tensor fields. More research into these pseudo-features is also necessary to study their significance.

4.3 Root Finding

In order to extract smooth and continuous feature lines in 3D tensor fields, we look at each of the 6 faces of every hexahedral cell. For each face, we extract the intersection point(s) of the feature lines. These points are then connected to generate a continuous feature line.

We know that the degenerate 3D tensors on a 2D slice are mostly points. The only exception is if the feature line lies exactly *on* the face. But even for that case, that feature line will intersect an adjacent non-coplanar face on the edge, or possibly corner vertex. To find these feature (intersection) points that satisfies all seven constraints simultaneously, we employ a modified version of Newton-Raphson algorithm to solve such a over-specified system of equations.


(a) Two degenerate points

(b) Three degenerate points

Figure 1: White dots are degenerate points indicating places where all seven constraint functions are zero. Each colored curve corresponds to a constraint function being equal to zero. Places where multiple curves intersect are where multiple constraint functions are satisfied simultaneously. The background is pseudo-colored by the discriminant functions. The data is a 2D slice of a randomly generated 3D tensor field.

Assume the tensor field is $T(X)$. For the feature points X^*, we have $\overrightarrow{CF}(X^*) = CF_i(X^*) = 0$, for $i = 1, ..., 7$, where $\overrightarrow{CF}(X)$ is an assembly of the seven constraint functions into one vector function. Using the modified Newton-Rhapson method and an initial guess of X_n, we have the following updating formula,

$$X_{n+1} = X_n - \left(\frac{\partial \overrightarrow{CF}}{\partial X}^T \cdot \frac{\partial \overrightarrow{CF}}{\partial X} \right)^{-1} \left(\frac{\partial \overrightarrow{CF}}{\partial X}^T \cdot \overrightarrow{CF} \right) \Bigg|_{X=X_n} \tag{10}$$

$$\frac{\partial \overrightarrow{CF}}{\partial X} = \frac{\partial \overrightarrow{CF}}{\partial T} \cdot \frac{\partial T}{\partial X} \tag{11}$$

Note that we calculate the $\frac{\partial \overrightarrow{CF}}{\partial X}$ from the chain rule using $\frac{\partial \overrightarrow{CF}}{\partial T}$ and $\frac{\partial T}{\partial X}$ rather than from the interpolated values of $\overrightarrow{CF}$ on the grid using finite difference methods for higher precision. $\frac{\partial \overrightarrow{CF}}{\partial T}$ is calculated from the formula of the tensor constraints, and $\frac{\partial T}{\partial X}$ is from the interpolated tensor values. We used both the bilinear and bicubic natural spline interpolations.

Using the center of each cell as the initial guess for an intersection point, we find that this method converges to the actual intersection point within five iterations in most non-degenerate cases with a precision up to 10^{-9}, and it almost never misses a feature point if it exists. Additional points are obtained by subdividing the cell face. This modified Newton-Rhapson method on constraint functions is superior in speed, accuracy and precision compared to other methods developed directly based on the cubic discriminants. For example, we also implemented a comparison algorithm based on cubic discriminant that searched for its minimum using conjugate gradient methods. Not only is it about 50 times slower, using any precision less than 10^{-6} will yield a false negative rate of over 50%.

4.4 Connecting Feature Points

Some cells may have more than one pair of intersection points and hence more than one feature line. We use a multi-pass approach to connect these intersection points. We only examine candidate cells that contain intersection points on at least one of their six faces. In the first pass, all candidate cells containing exactly two intersection points are processed by: (a) simply connecting those two points, (b) recording the orientation of the line segment as tangents at the end points, and (c) marking the cell as processed. In the subsequent passes, their unprocessed neighboring candidate cells are processed by connecting a line segment between each pair of intersection points in such a way as to minimize the angle deviation between the tangent recorded at the end point and the line towards other intersection points within the cell. Each neighboring candidate cell is marked as processed, and the procedure continues until there are no more candidate cells.

In our current implementation, we use this iterative method to generate the tangent lines on feature points and ultimately resolve the line connections between multiple points. In the future, we plan to calculate the tangent of the degenerate tensor line at a specific feature point analytically instead of from the post-processing method.

4.5 Categorization of Degenerate Lines

There are two cases where hyperstreamlines cross each other in double degenerate tensors. The first case (type P for planar) is when the major and medium eigenvalues are identical, and the second case (type L for linear) is when the medium and minor eigenvalues are identical. We define a quantity K, that measures the *eigen difference*, of a tensor T with eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$:

$$K = 2\lambda_2 - (\lambda_1 + \lambda_3) \tag{12}$$

The eigen difference K measures whether the tensor at a point is closer to type P or type L. It is easy to show that K returns a positive value for a type P degenerate tensor, and returns a negative value for a type L degenerate tensor. When K equals zero, the three eigenvalues are the same, and the tensor is triple degenerate. We color code the feature lines with their associated eigen differences: warm colors for type P and cool colors for type L. The closer the color is to pure green, the closer the tensor is to triple degeneracy (See Figures 2 - 4). Since triple degenerate points are the only locations where a type P and type L feature line can cross, coupled with the color mapping for K, the combined presentation provides a strong visual clue for finding triple degenerate points even though these are not explicitly calculated.

4.6 Higher Order Degeneracy

Previous researchers have pointed out that tensor features may have higher order degeneracy than just points, and may include lines, surfaces and subvolumes. It can be proven that not only is the discriminant zero at the degenerate tensors, but also the first order gradient of the discriminant is also equal to zero. We hypothesize that the higher order degeneracy happens only where the second or higher order of discriminant gradients are zeros. One can find this kind of phenomenon only under very special conditions. For example, we can show that for the single point load data, there are two types of degenerate features: one is a feature line directly below the point load direction; the other is a surface spreading symmetrically down from the point load. The higher order surface degeneracy happens in this particular dataset because of a delicate equilibrium achievable only in a purely synthetic data. This equilibrium can be easily disturbed by noise or other forces, and the feature will fall back to the more stable form of lines. This is confirmed in the double point load data. In this data, although the discriminants on the degenerate surface are still very small, as indicated by the weak transparent surface in Figure 4, the stable tensor features are lines.

In the same figure, we notice that near the apex of these two weak surfaces, the feature lines seem to break apart. We think that in the vicinity of those two points, the appropriate tensor feature is a small feature surface. Since our current algorithm is not designed for finding such features, it appears as an artifact. This is an area that we plan on addressing as well – even though such features are generally unstable.

Another generally unstable higher order feature is feature subvolumes. We think that in some type of tensor data sets, such as diffusion tensor data sets, it may be more likely to find such features. The locations of such features can be found in purely isotropic regions, and while they may not be of too much interest, we also plan to investigate this aspect, just for completeness.

5 Implementation Issues

In this section, we discuss two implementation issues affecting the speed and accuracy of the results.

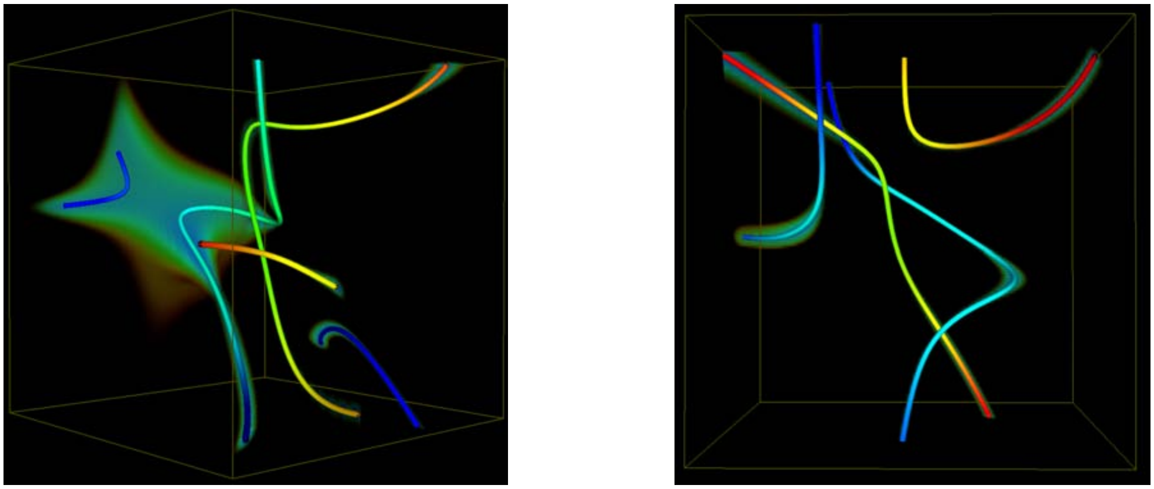
5.1 Pre-filtering

Methods based on *constraint functions* are superior over those directly based on *discriminants* because their signs could either be positive or negative around the features. This property allows us to detect sign changes in the vicinity of the tensor features without referring to the unpredictable gradients of tensors. This property also allows us to quickly eliminate those cells without sign changes from further consideration when finding tensor features within them, thereby speeding up the whole feature extraction process. The pre-filtering procedure proceeds by successively evaluating the seven constraint functions for each cell. If a cell shows that there is no sign changes from any single one of the seven constraint functions, and their values are significantly away from zero, then subsequent evaluations are skipped since the cell can be eliminated as a candidate cell.

5.2 Interpolation Methods

Usually the features lines are stable in non-degenerate data set. But in rare cases, such as when the features are higher order e.g. surface or subvolume features, the feature lines become unstable in that a small amount of noise could dramatically change the features. In our experiments, this occurs in one area of a synthetic data – the bifurcating branch below the loads in the double point load stress tensor. More details about this data set will be discussed in Section 6.

In this area, not only are the discriminants and their first order gradients zero, but the second order gradients are also very small. This higher order degeneracy makes degenerate tensor features extremely sensitive even to the small noise introduced by interpolation. Figure 3 shows the effects of varying the interpolation method. In both images, we first compute the tensor values on a 32^3 grid and apply our numerical method to extract the degenerate tensors with a specified interpolation method. To confirm the correctness of the results, we also resampled the tensor values from the original grid to a higher resolution grid of 128^3 using the same interpolation method and volume rendered it. The colors are mapped to the discriminant, where less transparent blue colors are closer to zero, and more transparent red colors have higher values. The features should be located in the blue regions. Since the implementation of the resampling and the volume rendering is straightforward, it also serves as a visual verifier for our numerical algorithm. Figure 3(a) uses linear interpolation, and the degenerate features lines appear broken and disorganized. In contrast, the result from tricubic natural spline interpolation is shown in Figure 3(b). We can see it fixed

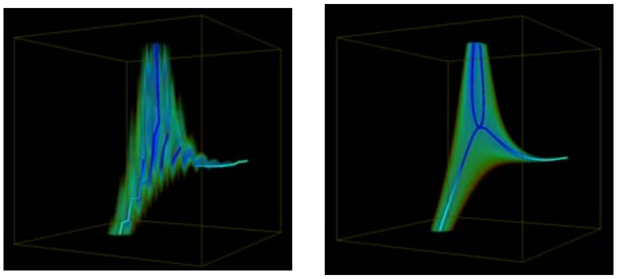


(a) First set

(b) Second set

Figure 2: Randomly generated 3D tensors. Warmer line colors are closer to type P degenerate points where major and medium hyperstreamlines intersect, while cooler line colors are closer to type L degenerate points where medium and minor hyperstreamlines intersect. The rest of the volume is pseudo-colored by the discriminant using cool colors for low discriminant values (closer to feature lines) and warm transparent colors for distant values.

the problem almost perfectly. Actually, we also tried other higher-order interpolation methods such as: tricubic, triquartic, triquintic and Hermite interpolation. They all have improved results over the trilinear interpolation, but do not address the problem as satisfactorily as the tricubic natural spline interpolation.



(a)

(b)

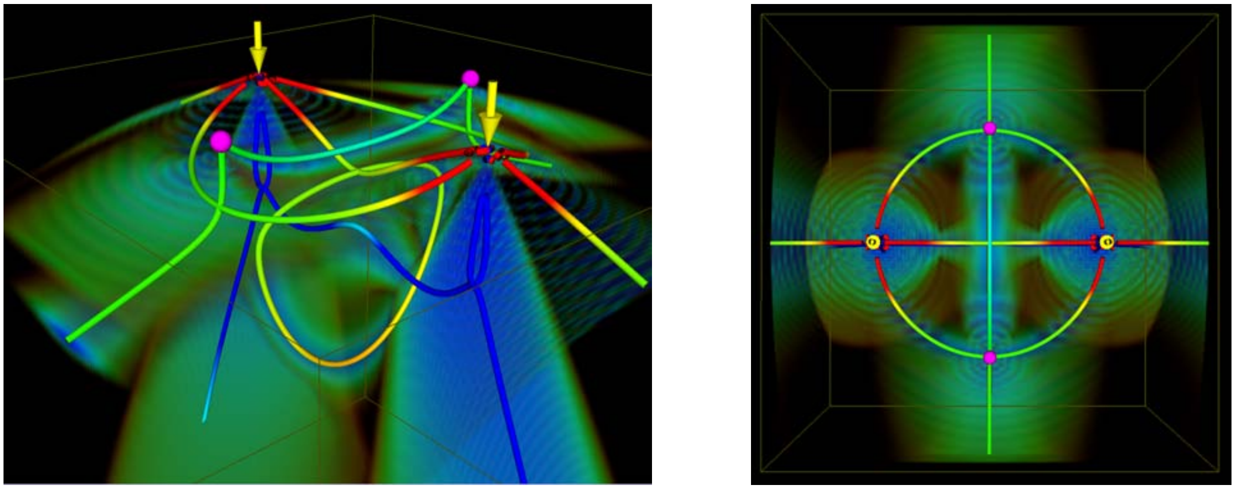
Figure 3: Closeup of the bifurcating feature lines below the point loads in the double point load data set. Feature lines are colored by the eigen difference while the volume is colored by the discriminant using cool colors for low discriminant values (closer to feature lines) and warm transparent colors for distant values. (a) Trilinear interpolation, (b) Tricubic natural spline interpolation.

6 Results

We experimented with four data sets to test out our degenerate tensor extraction algorithm. The first is a 2D rectangular patch with symmetric 3D tensors at the four corners that have been set randomly (see Figure 1). The tensor values within the patch are obtained through linear interpolation. This synthetic data corresponds to tensors on a face of a 3D cell. The second is a 3D cell with symmetric 3D tensors on its eight corners which are also set randomly (see Figure 2). It is sampled into a higher resolution for smoother features lines. The third is the stress tensor data in a semi-infinite volume with two point loads (see Figure 4). The fourth is the deformation tensors in the computed flow past a cylinder with hemispherical cap (see Figure 5). For Figures 2 to 4, the colors of the volumes are mapped to the tensor discriminant (Equation 6) with more transparent cooler colors mapped to lower values and more opaque warmer colors mapped to higher values. Degenerate tensors can be found in the cool blue regions. Digital images can be accessed online at: www.cse.ucsc.edu/research/avis/tensortopo.html.

Figure 2 shows degenerate tensors in a 3D cell form feature lines (rendered as tubes). Note that the feature lines are *not* hyperstreamlines, rather they are where the major and medium, or the medium and minor, or all three hyperstreamlines intersect each other. The color of the tubes are mapped to the eigen difference, where the type P lines are mapped to warmer colors and the type L lines are mapped to cooler colors. Only the faint green is visible in the vicinity of the tubes because the tubes are in the blue regions. We see that complex feature lines can form even from a simple linearly interpolated random tensor field. In (a), the type P and L lines swirl around each other, while in (b), the two types of lines form a complicated structure.

Figure 4 shows the double point load stress tensors. The yellow arrows indicate the two point loads, and the two magenta spheres are the triple degenerate points. We can see the line of double degeneracy connecting these two stress-free points as alluded to in [7]. Other very interesting feature lines are also extracted. The first is a vertical loop that lies directly under the double degenerate feature line connecting the two triple degenerate points. This feature is not present in the single point load data. This loop feature is also stable in the sense that it persists even as the magnitudes of the two point loads are varied. The second is how the blue feature line below each of the point load bifurcate and then reconnect. These two structures and the vertical loop are connected together by a type P feature line running between the two point loads. Looking from the top view in (b), we see another interesting feature which is the circular feature line that connects the two point loads and the two triple degenerate points. We need to further investigate the physical significance of these features that have not been seen in previous visualizations of the data. It is worth noting that the stress tensor is dominated by only one single load in the vicinity of the load point, so it is locally similar to the single point load stress tensor where the degenerate


(a) Oblique view

(b) Top view

Figure 4: Double point load data. Yellow arrows indicate point load, while the 2 magenta spheres show the location of the triple degenerate points. Feature lines are colored by the eigen difference while the volume is colored by the discriminant.

tensor form a surface symmetrically spreading away from the load point. Since our algorithm is designed for extracting features lines, it produces artifacts when the features form a surface or subvolume.

Figure 5 shows degenerate lines in the deformation tensors of the computed flow past a cylinder with a hemispherical cap. Only a portion of the data close to the cap is shown because most of the interesting features are found there. First, we see a curved line on the cap shown by the black arrow. It matches some of the patterns of the velocity topology from the same data set. There are more features at the upper half of the data because the flow there is more turbulent. Figure 5(a) is from a oblique view. Most of the features are close to the geometry of the object except a complicated branch structure which extends away from the geometry. It contains a small cyan horn shape indicated by the pink arrow, two green ring shapes, and a bifurcating structure. Figure 5(b) is from a top view. Again, we see that most of the interesting features are very close to the geometry of the object. We also see a strong type L (blue) bulb shape structure that extends to the end of the cylinder. Near the cap, it intertwines with a strong type P (red) heart shape structure. This phenomenon is interesting in that although these two structures are very close, they do not cross each other, because there is no green color at their visual cross points. It also means the tensors are quite turbulent in this area, because large variations are happening in very close proximity. There are many other interesting features shown in these two images. In the near future, we will investigate the significance of these features and finish a formal analysis on the role of the degenerate tensor lines in 3D tensor field.

7 Conclusion

We pointed out that the degenerate tensors form lines in 3D real symmetric tensor field. This knowledge lays a foundation for future research on topology based methods to visualize 3D tensor fields. Furthermore, we presented an algorithm to extract degenerate tensor lines. This algorithm uses a new formula that decomposes cubic discriminant into a sum of squares of seven polynomials. Feature points are first extracted on each face of a candidate hexahedral cell, and the points connected in an iterative fashion to generate feature lines. We applied this algorithm on several data sets including randomly generated tensor fields which allowed us to stress test our algorithm, several analytical data set e.g. single and double point load data sets to validate our results, and several computational data set e.g. flow past cylinder with hemispherical cap, to test its practical use. The results point to new knowledge, for the case of the double point load data set, as well as additional areas of investigation such as studying the correlation between the interesting patterns we saw in the real data sets and the underlying physics. These new insights will be useful in seeding hyperstreamlines, topology simplification, and tracking topology in time-varying data.

8 Acknowledgment

This work is supported by NSF ACI-9908881. The flow data is courtesy of NASA. We would also like to thank Peter Lax, Beresford Parlett, and Xavier Tricoche for their correspondence and suggestions. The comments by the reviewers also helped to improve this paper.

References

[1] A. Bhalerao and C.-F. Westin. Tensor splats: Visualising tensor fields by texture mapped volume rendering. In *Sixth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'03)*, pages 294–901, Montreal, Canada, November 2003.

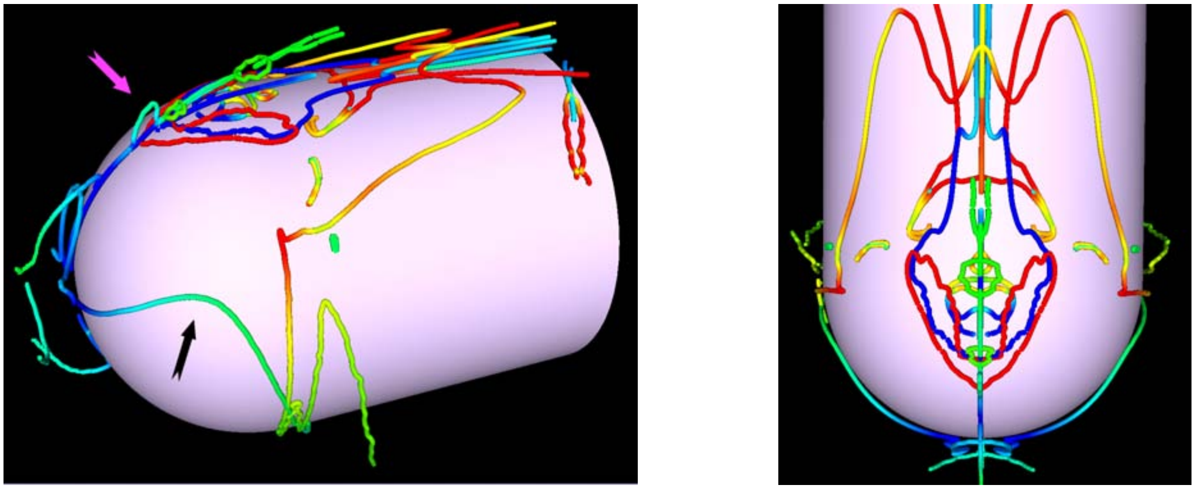
[2] E. Boring and A. Pang. Interactive deformations from tensor fields. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *Proceedings IEEE Visualization '98*, pages 297–304. IEEE Computer Society Press, 1998. Tensor / Flow.

[3] W.C. de Leeuw and J.J. van Wijk. A probe for local flow field visualization. In G.M Nielson and D. Bergeron, editors, *Proceedings IEEE Visualization '93*, pages 39–45. IEEE Computer Society Press, 1993.

[4] T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, July 1993.

[5] T. Delmarcelle and L. Hesselink. The topology of second-order tensor fields. In R.D. Bergeron and A.E. Kaufman, editors, *Proceedings IEEE Visualization '94*, pages 140–148. IEEE Computer Society Press, 1994. Flow Features and Topology.

[6] R.B. Haber. Visualization techniques for engineering mechanics. *Computing Systems in Engineering*, 1(1):37–50, 1990.



(a) Oblique View

(b) Top View

Figure 5: Degenerate lines in deformation tensors of flow past a cylinder with a hemispherical cap. Feature lines are colored by the eigen difference.

[7] L. Hesselink, Y. Levy, and Y. Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, Jan-Mar 1997.

[8] D. Hilbert. Ǔber die darstellung definiter formen als summen von quadraten. *Math. Annalen*, 32:342–350, 1888.

[9] Gordon Kindlmann. Superquadric tensor glyph. In *Vissym'04*, 2004. http://www.cs.utah.edu/˜gk/papers/vissym04/.

[10] Gordon L. Kindlmann and David M. Weinstein. Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *IEEE Visualization*, pages 183–189, 1999.

[11] R.M. Kirby, H. Marmanis, and D.H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In David Ebert, Markus Gross, and Bernd Hamann, editors, *Proceedings of Visualization 99*, pages 333–340, San Francisco, 1999.

[12] David Laidlaw, Eric Ahrens, David Kremers, Matthew Avalos, Russell Jacobs, and Carol Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings of Visualization 98*, pages 127–134, 1998.

[13] P. D. Lax. On the discriminant of real symmetric matrices. *Communications on Pure and Applied Mathematics*, LI:1387–1396, 1998.

[14] Peter Lax. *Linear Algebra*. Wiley, 1996.

[15] B. N. Parlett. The (matrix) discriminant as a determinant. *Linear Algebra and its Applications*, 355:85–101, 2002.

[16] X. Tricoche, G. Scheuermann, and H. Hagen. Tensor topology tracking: A visualization method for time-dependent 2D symmetric tensor fields, 2001.

[17] Xavier Tricoche, Gerik Scheuermannn, and Hans Hagen. Topology simplification of symmetric, second order 2D tensor fields. pages 275–292, 2003.

[18] D.M. Weinstein, G.L Kindlmann, and E.C. Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In D. Ebert, M. Gross, and B. Hamann, editors, *Proceedings IEEE Visualization '99*, pages 249–254. IEEE Computer Society Press, 1999. Vector and Tensor Visualization.

[19] Xiaoqiang Zheng and Alex Pang. Volume deformation for tensor visualization. In *Proceedings of Visualization 02*, pages 379–386, Boston, 2002.

[20] Xiaoqiang Zheng and Alex Pang. HyperLIC. In *Proceedings of Visualization 03*, pages 249–256, Seattle, 2003.

[21] Xiaoqiang Zheng and Alex Pang. Interaction of light and tensor fields. In *VisSym'03*, pages 157–166, 295, May 2003. www.cse.ucsc.edu/research/avis/tensorray.html.

[22] Xiaoqiang Zheng, Xavier Tricoche, and Alex Pang. Degenerate 3D tensors. Technical Report UCSC-CRL-04-09, University of California, Santa Cruz, 2004.